



Tecnológico de Monterrey

M2. Tarea - Introducción a Unity

Manuel Villalpando Linares A01352033

13 de agosto del 2023

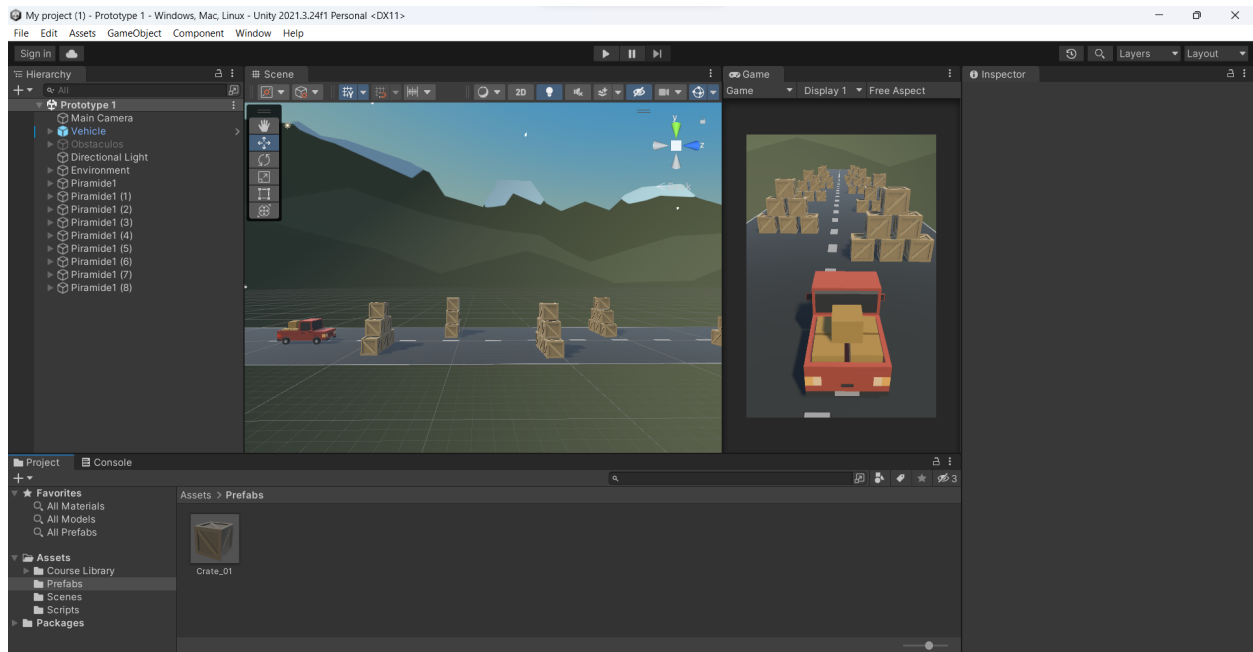
Modelación de sistemas multiagentes con gráficas computacionales
(Gpo 102)

Denisse Lizbeth Maldonado Flores

Alejandro Fernández

Laboratorio 1-1

- Screenshot del layout seleccionado y una justificación de por que lo seleccionaste.



Seleccioné el layout casi que tenía unity por default, ya que me permite hacer mis tareas de manera ordenada, siempre teniendo en cuenta en dónde está cada elemento que lo compone, es decir, sin perderlo de vista. Además, esto me permite tener mejor control en cuanto al inspector, ya que puedo observar más de los componentes que complementan a algún objeto principalmente para identificar cuales tengo y si me falta alguno. La barra de jerarquía izquierda de igual manera me permite observar más cómodo todos los objetos que tengo dentro del juego. Además de esto le puse la ventana de la cámara principal dentro del juego para ir imaginando cómo se vería una vez el usuario empieza a jugar. La ventana de la escena es la que ocupa el espacio principal ya que es en la que se trabaja principalmente, sin embargo esta se corta de abajo debido a que considero igual de importante el explorador de carpetas para ir agregando lo que me haga falta.

- Reflexión sobre el aprendizaje adquirido en el laboratorio 1-1 con fecha de realización.

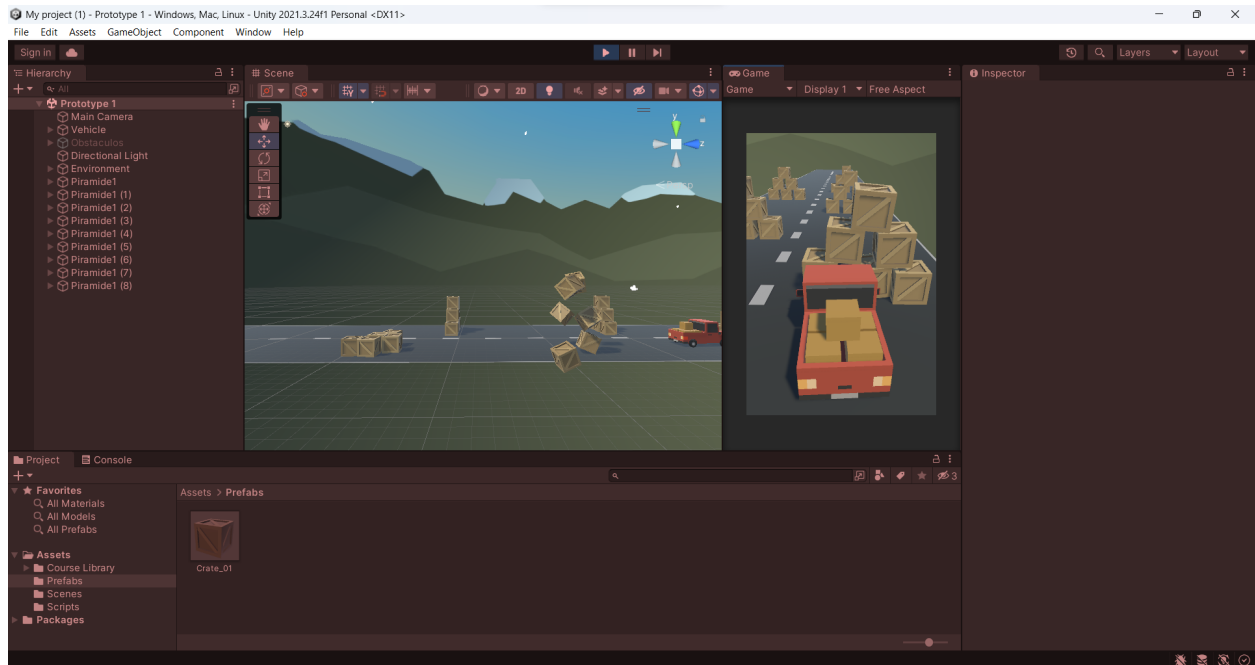
Al empezar en el mundo de Unity, descubrí que cada objeto es clave. Moviendo y explorando, vi que todo tiene límites, hasta el espacio virtual de momento, ya que vamos empezando, pero aún así se pueden crear cosas muy grandes. Mi carrito, anduvo esquivando y chocando entre obstáculos mientras ajustaba la cámara. Colocar objetos estratégicos y programar el movimiento

del carrito fue lo que más me gustó porque me hizo imaginar cómo es que se vería dentro del juego como tal.

Fecha de realización 12/08/2023.

Laboratorio 1-2

- Screenshot de tu layout de color en el play mode.



- Lista de errores que hasta el momento existen en el juego.

1. El carro no avanza según las ruedas.
2. Cuando das reversa el carro se mueve como una persona, es decir, no se ve afectado por la rotación en ángulo que se supone deberían de tener las ruedas. Es decir, se mueve como persona no como carro porque las llantas delanteras no dan la dirección, sino que la toma en conjunto.
3. El carro puede volar
4. El carro puede avanzar estando de cabeza o de lado.
5. Se buguean las cajas.

- Código documentado de la clase ****FollowPlayer****.

// Este script controla el comportamiento de una cámara que sigue al jugador en el juego.

// La cámara se posiciona con un desplazamiento predefinido con respecto al jugador.

using UnityEngine;

// La clase FollowPlayer se usa para seguir al jugador con la cámara.

// MonoBehaviour es una clase de Unity y se utiliza para crear scripts que interactúan con los componentes

```
public class FollowPlayer : MonoBehaviour
```

```
{
```

```
    public GameObject player; // Referencia al GameObject del jugador que esta cámara seguirá.
```

```
    private Vector3 offset = new Vector3(0, 6, -7); // Desplazamiento para controlar la posición de la cámara en relación al jugador.
```

// El método Start se llama antes del primer frame de actualización

```
void Start()
```

```
{
```

```
    // Este método se usa típicamente para tareas de inicialización.
```

```
    // No se necesita inicialización específica aquí.
```

```
}
```

// El método LateUpdate se llama después de Update, asegurando que el movimiento del jugador haya sido procesado.

```
void LateUpdate()
```

```
{
```

```
    // Calcula la posición deseada para la cámara usando la posición del jugador y el desplazamiento.
```

```
    Vector3 posicionDeseada = player.transform.position + offset;
```

```
    // Actualiza la posición de la cámara para que coincida con la posición deseada.
```

```
    transform.position = posicionDeseada;
```

// En el método LateUpdate, es común realizar tareas relacionadas con la cámara después de aplicar el movimiento del jugador.

```
// Esto evita temblores en la cámara y garantiza un seguimiento de la cámara más suave.  
}  
}
```

- Reflexión sobre el aprendizaje adquirido en el laboratorio 1-2 con fecha de realización.

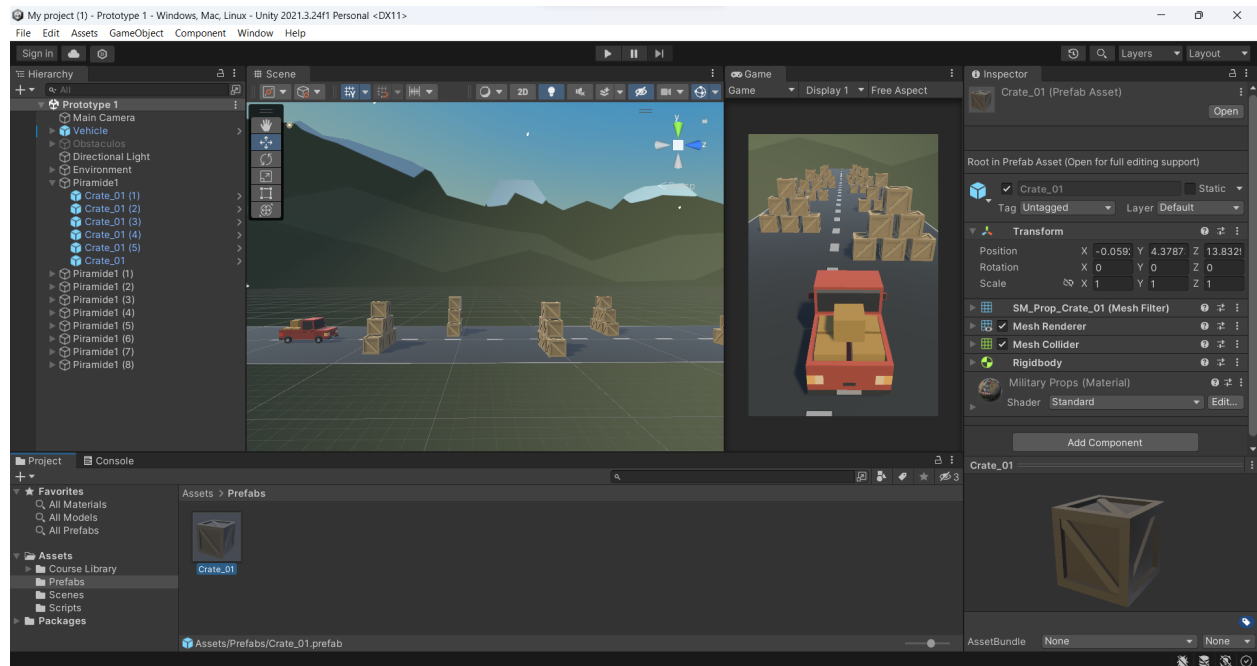
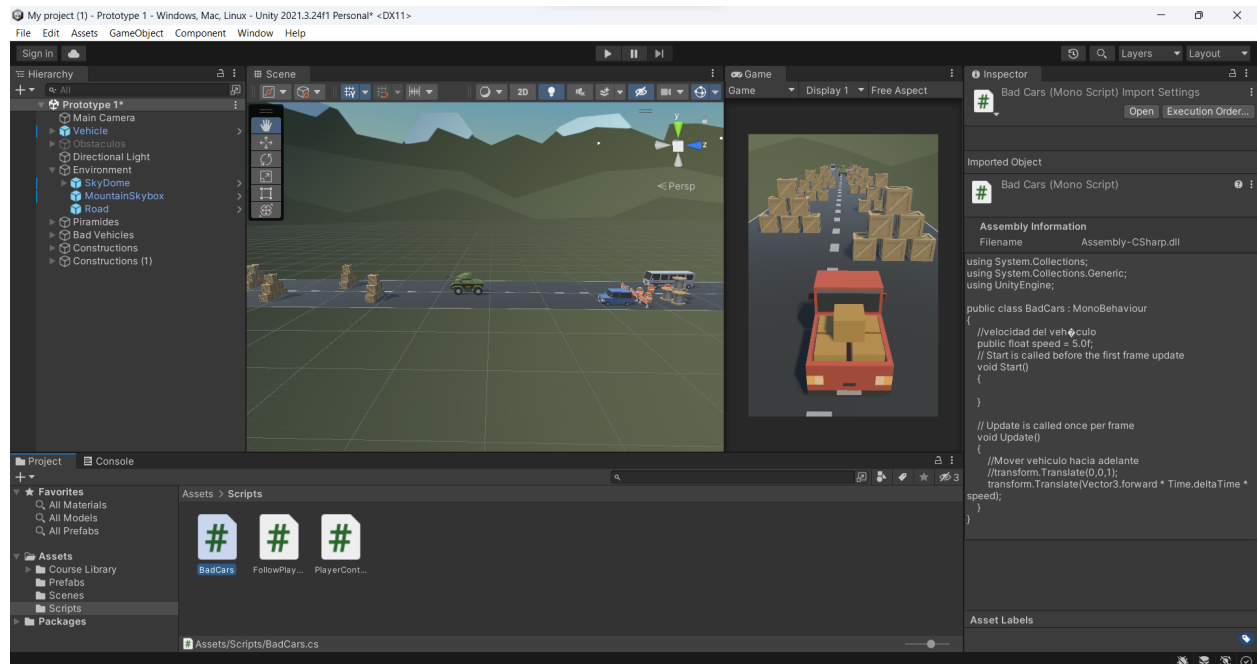
Comprendí cómo implementar físicas y obstáculos de manera efectiva. Ajusté la velocidad usando "Time.deltaTime" para movimientos realistas, agregué físicas a objetos con "Rigidbody" y ajusté propiedades para que el juego tuviera interacciones entre objetos más naturales. Usé duplicación para crear obstáculos tales como las pirámides además de que aprendí a agrupar objetos dentro de un objeto padre, esto para poder mostrar u ocultar distintos "Presets" de obstáculos dentro del mapa, lo cual personalmente fue lo que más me gustó ya que se me hizo muy interesante porque no lo sabía, además de que es muy útil, seguido de un script para mejorar la cámara. Cambié el input en el "Input Manager" y ajusté el movimiento para mejorarlo.

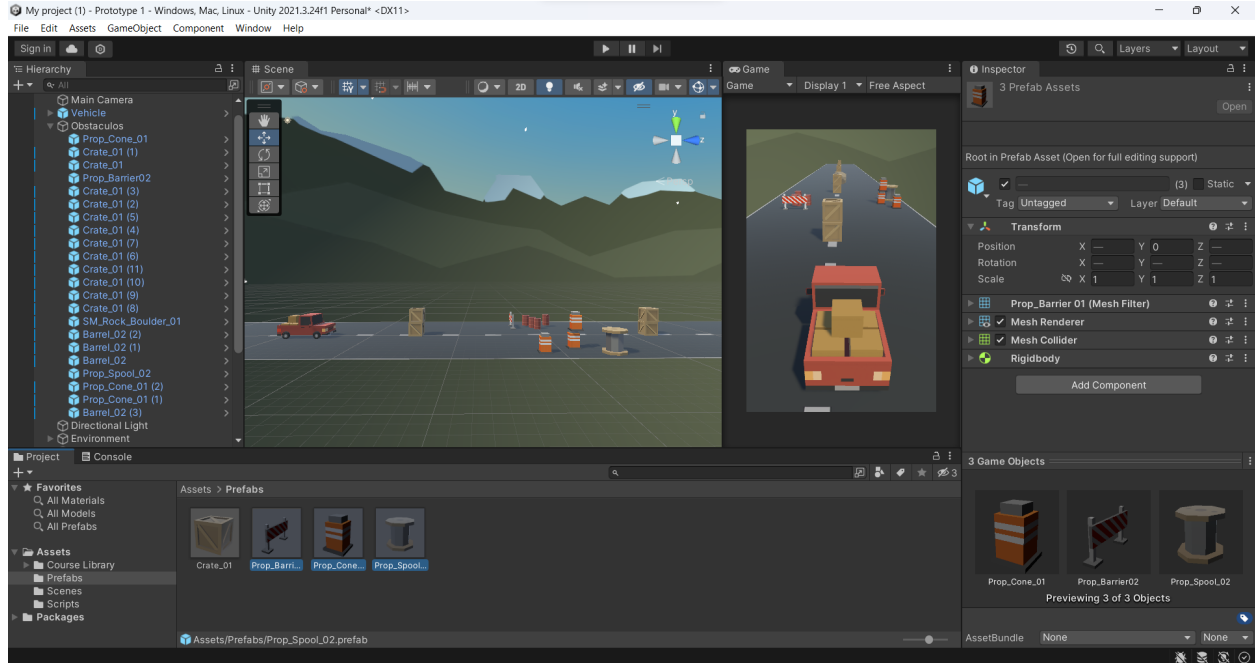
También organicé el código y entendí la importancia de la estructura en Unity.

Fecha de creación 12/08/2023.

Laboratorio 1-3

- Listado con explicación y screenshots de los prefabs de obstáculos que hayas realizado.





- Create_01: Caja cúbica importada de los objetos por default de Unity con rigidbody integrado para que pueda tomarse como objeto con el que se puede interactuar, y agrega físicas por medio de la misma librería.
 - Prop_Barrier: Barrera de tránsito igual con rigidbody de la librería de objetos predeterminada de Unity que cree para un set de obstáculos de un curso de Unity en coursera aparte.
 - Prop_Cone: Cono vial, de igual manera con rigidbody con una masa extremadamente ligera, creado para otro set de obstáculos.
 - Prop_Spool: Bobina creada como obstáculo con una masa extremadamente pesada, de igual manera para otro set, esto para ver posibles glitches del carro chocando con objetos más pesados que sí mismo.
 - BadCars: Agregue carros al otro extremo del mapa, además de un obstaculo extra, los carros avanzan automáticamente y pretenden ser obstáculos extra para impedir el paso del jugador.
- Liga al video con el resultado final de tu proyecto y el nivel diseñado.**

<https://youtu.be/NOII3glyu7o>

- Reflexión sobre el aprendizaje adquirido en el laboratorio 1-3 del con fecha de realización.

En el último laboratorio, exploramos cómo usar prefabs y crear un juego multiplayer en local desde nuestra misma laptop. Creamos objetos personalizados con prefabs y configuramos propiedades. Personalmente me tomé la libertad de agregar tres vehículos más al mapa solo para hacerlo más divertido y explorar un poquito más el software. Aprendimos a posicionar elementos y usar cámaras para diferentes perspectivas. Mediante Input IDs, permitimos control independiente a ambos jugadores. Así evitamos duplicación de código, pudiendo hacer más fácil de leer y darle mantenimiento. Finalmente, creamos un nivel con obstáculos y grabamos un video para mostrar el resultado lo cual fue mi parte favorita ya que pude personalmente probar el juego y explorar lo que se podía hacer en él. Ganamos habilidades en diseño, scripting y configuración básica para correr juegos en multiplayer local en Unity.